AF/2175 120

#9
7/28/04 AW

PATENT
Atty. Dkt. No. ROC920010073US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Driesch et al.

Serial No.: 09/970,352

Confirmation No.: 1536

Filed: October 3, 2001

For: Reduce Database Monitor Workload by Employing Predictive Query Threshold

Group Art Unit: 2175

Examiner: Rones, C.

MAIL STOP Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING
37 CFR 1.8
I hereby certify that this correspondence is being deposited on July 16, 2004 with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450 Alexandria, VA 22313-1450.

July 16, 2004
Date

Gero G. McClellan

Dear Sir:

APPEAL BRIEF

Appellants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2175 dated February 10, 2004, finally rejecting claims 1-29. Please charge the fee of $330.00 for filing this brief to Deposit Account No. 09-0465. **Three (3) copies of this brief are submitted for use by the Board.**

Real Party in Interest

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

Related Appeals and Interferences

Appellant asserts that no other appeals or interferences are known to the Appellant, the Appellant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-29 are pending in the application. Claims 1-27 were originally presented in the application. Claims 28-29 were added in the Response to Office Action dated October 3, 2003. Claims 1-29 stand rejected in view of a singular reference as discussed below. The rejection of claims 1-29 based on the cited reference is appealed. The pending claims are shown in the attached Appendix.

Status of Amendments

One amendment to the claims was submitted after the final rejection and entered by the Examiner. Arguments presented after final rejection were not accepted by the Examiner.

Summary of Invention

The present invention provides a method of selectively logging query implementation information of a query in a database management system (pg. 5-6, para. 0017). The method comprises determining an execution plan for the query (pg. 8, para. 0028-0029), determining whether query implementation information should be logged for the query (pg. 9, para. 0031, pg. 3, para. 0004-0005), and executing the query (pg. 9-10, para. 0032).

Issues Presented

1. Whether the Examiner erred in rejecting claims 1-29 under 35 U.S.C. 102(e) as being anticipated by *Carino, Jr.* (U.S. Pat. No. 6,353,818, hereinafter *Carino*).

Grouping of Claims

Pending claims 1-29 stand or fall together for all arguments presented by Appellants and claim 1 is representative of the claims.

ARGUMENT

I. THE EXAMINER ERRED IN REJECTING CLAIMS 1-29 UNDER 35 U.S.C. § 102(e) BECAUSE *CARINO* FAILS TO TEACH, SHOW, OR SUGGEST SELECTIVELY LOGGING QUERY IMPLEMENTATION INFORMATION OF A QUERY.

Claims 1-29 stand rejected under 35 U.S.C. § 102(e) as being anticipated by *Carino, Jr.* (U.S. Pat. No. 6,353,818, hereinafter *Carino*) on grounds that *Carino's* method encompasses or overlaps the instantly claimed method. Appellants respectfully traverse this rejection.

The present claims disclose a method for selectively logging query implementation information (pg. 5-6, para. 0017). The term "query implementation information" is self-describing and refers to information pertaining to the implementation of a particular query. The method comprises receiving a query, determining an execution plan for the query, determining whether query implementation information should be logged for the query, and executing the query (*See, e.g.,* Claim 1). Thus, the present claims describe a query, an execution plan for a query, and query implementation information which is selectively logged.

A rejection under 35 U.S.C. § 102(e) requires that <u>each</u> element be disclosed by the reference. *See* MPEP § 2131. Respectfully, the Examiner's rejection fails to satisfy this statutory requirement. The Examiner states that *Carino* discloses determining whether query implementation information should be logged for a query, citing the Abstract, Fig. 2, and Col. 6, Lines 24-67 (Paper No. 4, Page 2). In point of fact, *Carino* makes no such disclosure. *Carino* describes method of executing a database query in a database management system (*See* Claim 1). More specifically, *Carino* is directed to query optimization in which a query plan generator generates several query plans, one of which is chosen by a query plan evaluator (*See* Abstract). *Carino* does not disclose determining whether query implementation information should be logged for the query. Thus, Appellants respectfully submit that the rejection under 35 U.S.C. § 102(e) is improper because *Carino* does not teach each element of the present claims. Each of the specific portions of *Carino* relied on by the Examiner are now discussed.

In the Abstract, *Carino* describes creating query plans (analogous to execution plans for a query) based on a query. Upon receiving a query, a query plan generator generates query plans for the query, and a query plan evaluator selects a query plan according to a measured resource metric (*See* Abstract). Thus, the Abstract refers only to a query and selecting from query plans. The Abstract does not disclose determining whether query implementation information should be logged.

With reference to Fig. 2, the Examiner specifically cites Col. 6, Lines 24-67. However, neither Fig. 2 generally, nor Col. 6, Lines 24-67 specifically disclose determining whether query implementation information should be logged, nor is any query-specific tracking performed.

Fig. 2 shows the operation of the query plan optimizer and query plan evaluator (*See* Brief Description of the Drawings). Central to *Carino* is determining the cost of performing expensive or substantially variant functions on large data objects (*See*, Title and Col. 2, Lines 30-67). Of particular concern to *Carino* are user-defined functions (UDFs) (*Id.*). A UDF is a function that acts as a query predicate for large objects (LOBs) (Col. 2, Lines 33-36). As part of its solution to this problem *Carino* teaches a resource requirement predictor (Item 202, Fig. 2) that predicts the resource requirements for the database query, reads statistics/demographics (Item 208, Fig. 2), and determines if the database query from the user implicates the invocation of expensive or substantially variant UDFs (Col. 6, Lines 25-29). *Carino* further teaches that the "statistics/demographics can also be augmented with information from a historical usage of the UDF or the object" (Col. 6, Lines 34-36). The statistics/demographics (Item 208, Fig. 2) may be collected by tracking various metrics, such as a CPU metric, a memory metric, a disk metric, and a network metric (Fig. 2, Col. 6, lines 45-49). In other words, *Carino* teaches tracking metrics that are indicative of resource consumption of the UDFs.

The Examiner provides no analysis describing how this portion of *Carino* is believed to anticipate the claims but appears to suggest that tracking these UDF-related metrics is the same as determining whether to log query implementation information for a query. Respectfully, the Examiner errs in this regard. First, *Carino* does not disclose selective tracking of any information. Thus, even assuming that the UDF-related metrics (i.e., the statistics and demographics (Item 208, Fig. 2)) can be considered to be "query implementation information", these metrics are not selectively tracked. In fact, it stands to reason that the statistics and demographics (Item 208, Fig. 2) would always be tracked. The statistics and demographics (Item 208, Fig. 2) are essential to the operation of the system and method disclosed by *Carino*. Specifically, optimization of queries including UDFs necessarily requires using statistical/demographic data. (*See*, Col. 8, Lines 20-27.) It follows that the statistical/demographic data should be collected and updated on an ongoing basis in order to have good data on which to make the resource predictions. Thus, *Carino* does not explicitly or inherently disclose or suggest that data of any kind is selectively collected; rather any data collection is done indiscriminately without a prior determination of whether to collect the data. Second, *Carino* does not disclose selective tracking on a query level. Claim 1, for example, recites "determining whether query implementation information should be logged <u>for the query</u>." (Emphasis added.) That is, the determination of whether to log of the information is made with respect to a given query. Any tracking done by *Carino* is indiscriminate (i.e., done for all queries), and not selective. Even assuming that the function responsible for tracking the statistical/demographic data could be turned ON and OFF, a determination of whether to turn the function ON or OFF is not made for a given query, as claimed. Accordingly, *Carino* does not disclose selective logging of query implementation information nor does *Carino* disclose determining whether query implementation information should be logged for a query.

In Examiner's *Response to Arguments* (Paper No. 4, Page 6), the Examiner cites *Carino* at Col. 8, lines 43-67 and Col. 9, Lines 1-31 for the proposition that *Carino* discloses determining whether query implementation information should be logged for a query. The cited section discloses the process for selecting from query plans. A query plan is selected based on resource metrics (Col. 8, Lines 55-60). Resource metrics are specific to the available system resources, and do not refer to a specific query (Col. 9, Lines 2-10). Thus, the section cited in Examiner's *Response to Arguments* section does not disclose determining whether query implementation information should be logged for a query.

As a further note, Examiner's *Response to Arguments* (Paper No. 4, Page 6) states that *Chaudhri* discloses "keeping the measured values of a query". However, in the Final Office Action (Paper No. 4) being appealed from, claims 1-29 are only rejected under 35 U.S.C. § 102(e) as being anticipated by *Carino*. Accordingly, Examiner has not cited any statutory grounds under which *Chaudhri* may be relied upon. Thus, Appellants respectfully submit that Examiner's reliance on *Chaudhri* is improper. Further, Appellants respectfully submit that the final portion of Examiner's *Response to Arguments* contains sentence structure errors and rendering the language unclear. As such, it is not clear to the Appellant whether the Examiner inadvertently or purposely referred to *Chaudhri*. In any case, since *Chaudhri* was not relied on as a basis for rejection in the Final Office Action (Paper No. 4), any reference to *Chaudhri* is improper.

Conclusion

In conclusion, *Carino* does not disclose selectively logging query implementation information nor does *Carino* disclose determining whether query implementation information should be logged for a query. Thus, the rejection under 35 U.S.C. § 102(e) is improper because *Carino* does not teach each element of the present claims. The Examiner erred in finding that the method in *Carino* anticipates the instantly claimed method, system and article of manufacture. Thus, Appellants respectfully request reversal of the rejection of the claims.

Respectfully submitted,

Gero G. McClellan
Registration No. 44,227
MOSER, PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Agent for Appellants(s)

APPENDIX

1. (Original) A method of selectively logging query implementation information of a query in a database management system, the method comprising:

receiving a query;

determining an execution plan for the query;

determining whether query implementation information should be logged for the query; and

executing the query.

2. (Original) The method of claim 1, wherein determining whether query implementation information should be logged comprises determining whether the query should be monitored.

3. (Original) The method of claim 1, wherein determining whether query implementation information should be logged comprises comparing an estimated runtime of the query to a user-specified threshold value.

4. (Original) The method of claim 1, wherein determining an execution plan comprises determining a cost for a plurality of execution plans using one or more cost factors and wherein determining whether query implementation information should be logged comprises comparing at least one of the cost factors to a user-specified threshold value.

5. (Original) The method of claim 1, wherein determining whether the query implementation information should be logged comprises comparing a cost factor to a predefined threshold value.

6. (Original) The method of claim 5, wherein the cost factor is selected from at least one of a location of tables, a size of tables, a network node location, a system

operating characteristic, a system operating statistic, an estimated runtime for the query, space usage and any combination thereof.

7. (Original) The method of claim 1, wherein query implementation information is monitored by a monitor process and wherein determining whether the query implementation information should be logged is performed after executing the query.

8. (Original) The method of claim 7, wherein determining whether the query implementation information should be logged comprises determining whether a post-runtime cost factor exceeds a predefined threshold value.

9. (Original) The method of claim 1, wherein determining whether query implementation information should be logged comprises:

comparing a cost factor of the query to a threshold value; and

if the cost factor exceeds the threshold value, then logging the query implementation information.

10. (Original) The method of claim 9, wherein the cost factor is an estimated runtime of the query.

11. (Original) A computer readable medium containing a program which, when executed by a computer, performs an operation of selectively logging query implementation information of a query in a database management system, the operation comprising:

determining an execution plan for a query; and

determining, according to at least one threshold value, whether query implementation information should be logged for the query.

12. (Original) The computer readable medium of claim 10, further comprising executing the query.

13. (Original) The computer readable medium of claim 12, wherein determining whether query implementation information should be logged for the query is performed after executing the query.

14. (Original) The computer readable medium of claim 12, wherein the query is monitored by a monitor process during execution of the query and wherein determining whether the query implementation information should be logged is performed after executing the query.

15. (Original) The computer readable medium of claim 10, wherein determining whether query implementation information should be logged comprises determining whether the query should be monitored.

16. (Original) The computer readable medium of claim 10, wherein determining an execution plan comprises determining a cost for a plurality of execution plans using one or more cost factors and wherein determining whether query implementation information should be logged comprises comparing at least one of the cost factors to a threshold value.

17. (Original) The computer readable medium of claim 10, wherein determining whether query implementation information should be logged comprises comparing a cost factor to a predefined threshold value.

18. (Original) The computer readable medium of claim 17, wherein the cost factor is an estimated runtime of the query.

19. (Original) The computer readable medium of claim 17, wherein the cost factor is selected from at least one of a location of tables, a size of tables, a network node location, a system operating characteristic, a system operating statistic, an estimated runtime for the query, space usage and any combination thereof.

20. (Original) The computer readable medium of claim 10, wherein determining whether query implementation information should be logged comprises:

comparing a cost factor of the query to a threshold value; and

if the cost factor exceeds the threshold value, then logging the query implementation information.

21. (Original) The computer readable medium of claim 20, wherein the cost factor is an estimated runtime of the query.

22. (Previously Presented) A database system, comprising:

at least one database;

a query processor to generate execution plans for queries requesting information contained in the database;

a database engine to access the database according to the execution plans;

a threshold data structure containing at least one threshold value; and

a query implementation information log; wherein query implementation information for a query is written to the query implementation information log only if the threshold value is exceeded by a selected cost factor of the query.

23. (Original) The database system of claim 22, wherein the selected cost factor is an estimated runtime of the query.

24. (Original) The database system of claim 22, wherein the cost factor is selected from at least one of a location of tables, a size of tables, a network node location, a system operating characteristic, a system operating statistic, an estimated runtime for the query, space usage and any combination thereof.

25. (Original) The database system of claim 22, further comprising a monitor program which, when executed, monitors the query to collect the query implementation information.

26. (Original) The database system of claim 25, wherein the monitor program is executed only if the threshold value is exceeded by the selected cost factor of the query.

27. (Original) The database system of claim 26, wherein the selected cost factor is an estimated runtime of the query.

28. (Previously Presented) The method of claim 1, wherein determining whether query implementation information should be logged is done on the basis of the execution plan.

29. (Previously Presented) The computer readable medium of claim 12, wherein determining whether query implementation information should be logged for the query is done on the basis of the execution plan.